4

Domain Modeling for Language Analysis

Ralph Grishman

PROTEUS Project Memorandum #14
February 1988

prepared for the 1988 Duisburg Symposium
Linguistic Approaches to Artificial Intelligence
Universität Duisburg, March 23-26, 1988

This report is based upon work supported by the Defense Advanced Research Projects Agency under Contract N00014-85-K-0163 from the Office of Naval Research, and by the National Science Foundation under grant DCR-85-01843.

Domain Modeling for Language Analysis

Ralph Grishman
Department of Computer Science
New York University

prepared for the 1988 Duisburg Symposium
Linguistic Approaches to Artificial Intelligence

1. Introduction

It is taken for granted by now that a full understanding of natural language texts requires access to a rich store of world knowledge. Only through such world knowledge, in particular, can we determine discourse relations such as anaphoric reference and implicit temporal and causal relations between events. The crucial questions are: what information is required for natural language analysis, how should this information be organized, and how should it be utilized by the language analyzer?

While there has been considerable discussion and speculation on these issues, there have been relatively few studies involving complex, pre-existing texts. Our own study has involved *equipment failure messages* called CASREPs, which are prepared on board ships of the U. S. Navy. These messages describe the failure, diagnosis, and attempted repair of various types of shipboard equipment. In order to limit the domain modeling task, we have initially restricted ourselves to messages describing the failure of a single piece of equipment -- the starting air system for gas turbine propulsion systems. A typical message is

> DURING NORMAL START CYCLE OF 1A GAS TURBINE, APPROX 90 SEC AFTER CLUTCH ENGAGEMENT, LOW LUBE OIL AND FAIL TO ENGAGE ALARM WERE RECEIVED ON THE ACC [auxilliary control console]. (ALL CONDITIONS WERE NORMAL INITIALLY.) SAC [starting air compressor] WAS REMOVED AND METAL CHUNKS WERE FOUND IN OIL PAN. LUBE OIL PUMP WAS REMOVED AND WAS FOUND TO BE SEIZED. DRIVEN GEAR WAS SHEARED ON PUMP SHAFT.

The message analysis task is a rich one for exploring issues of knowledge representation and use. The equipment we have chosen is moderately complex, with several hundred functioning parts and involved interactions between the parts. The messages reflect this complexity in the variety of actions and of implicit temporal and causal relations which they convey. At the same time, the choice of a domain involving inanimate, manufactured objects offers certain advantages over other domains such as human social interactions. The domain is well-structured, interactions are well-defined, the system is mostly deterministic, and it is fairly well documented. As a result, it has been largely possible to delimit the domain knowledge required[1], and to produce a domain model whose accuracy can be verified independently of the texts to be analyzed.

In section 2 of this paper we briefly characterize our notion of understanding a text. In section 3 we give an overview of the system we have constructed for analyzing equipment failure messages, and indicate the points at which it makes use of domain information. We then turn in section 4 to the domain model itself, and describe how it provides the information needed by language analysis. We close with brief sections relating our work to other work on discourse analysis and discussing how our system's coverage may be broadened.

[1] We have limited the domain knowledge to logical structure and function, which is reasonably well-defined. However, a few messages involve the interaction of functional with spatial and material properties; the knowledge concerning these interactions is less clearly delimited and much more difficult to capture.

2. Text Understanding

A crucial characteristic of natural language is that so much of what is to be conveyed by a discourse remains implicit. The writer of a text expects that the reader shares a large body of world knowledge, and so he need only communicate a skeleton of facts, confident that the reader will flesh it out by inferring the connections between these facts. For example, if we read

Willa was hungry. She grabbed the Michelin guide.

(an example taken from (Wilensky 1981)) we recognize that these are not unrelated events, but rather that Willa wanted to find in the Michelin guide a good place to eat. There is an extensive literature on discourse coherence relations and the role of scripts, plans, and other world knowledge in understanding discourse (for example, (Schank and Abelson 1977), (Charniak 1972, 1978), (Hobbs 1982)).

The same phenomenon arises in equipment failure messages. If you hear

The engine won't start. The battery fluid is low.

you probably don't need to be told that the low fluid caused the engine failure, and not *vice versa*. We can also see from this example why establishing causal links is so crucial to understanding these messages: it allows us to distinguish cause from effect among the events mentioned in the message. By doing so, it tells us that the proper response is to add battery fluid, not to replace the engine.

Understanding a message also entails understanding the referents of the noun phrases in the message. There might be several batteries in your car (perhaps one in the clock, one in the stereo); you cannot claim to have properly understood the message without knowing which one of these batteries is being referred to.

If a message relates two or more events, we will also want to determine their time relationship:

John's fender was dented. He ran into a telephone pole.

As we see from this example, causal and temporal inference are tightly interlinked: because we recognize that running into a pole can dent your fender, we infer that his fender was dented immediately upon hitting the tree.

Although there certainly are other relations at work in these messages, these three -- reference (and coreference), causal relations, and temporal relations -- are probably most crucial to a proper understanding of the equipment failure messages. In the next two sections we consider how the language analyzer and the domain knowledge work together to establish these relationships.

3. The language analyzer

In order to test our conjectures regarding text analysis and the organization of domain knowledge, we have been building over the last few years a text understanding system named PROTEUS (PROtotype TExt Understanding System) and applying it to CASREPs describing the failure of a starting air system. All of the system components described below have been implemented except for the temporal analysis, whose design is still being elaborated. The system has been organized as a series of stages which process the information sequentially, largely in order to simplify system development. There are four major stages: syntactic analysis, clause semantics, noun phrase semantics, and discourse analysis (Figure 1). We shall briefly describe each stage in turn, dwelling longer on those stages which rely on the domain model.

3.1. Syntactic analysis

For syntactic analysis we have developed a sublanguage English grammar which covers the syntactic constructs encountered in these messages. It is in the form of an augmented context-free grammar: a context-free grammar supplemented by procedural *restrictions* which enforce various grammatical constraints (number agreement, subcategorization, etc.). The grammar is generally based on Harris's linguistic string theory, as adapted by Sager (1981) for automatic text analysis; the restrictions are written in a simplified version of the Restriction Language used earlier in the NYU Linguistic String Parser (Sager and Grishman 1975). Parsing is performed by a chart parsing algorithm (Thompson and Ritchie 1984).

Parsing is coupled with a procedure for syntactic regularization, whose main function is to reduce all the different forms of clauses (active and passive clauses, relative clauses, clauses with perfect and progressive tenses, some sentence fragments) into a canonical form consisting of a verb and syntactic-case-marked arguments. The regularized form is computed compositionally, working upwards from the bottom of the parse tree, by regularization rules which are associated with each production in the grammar.

3.2. Predicate semantics

The next stage, predicate semantics, transforms each clause into some combination of domain-specific predicates with arguments. Noun phrases whose heads are nominalizations are similarly translated into predicate-argument structures. Arguments are labeled with semantic case markers such as *agent*, *patient*, and *instrument*. Noun phrase arguments (other than nominalizations) are passed through unchanged.

This stage is based on the Inference Directed Semantic Analysis procedure developed by Palmer [Palmer 1983]. It uses three types of domain-specific information:

(1) a classification of the nouns of the domain into *semantic classes*

(2) predicate mapping rules, which specify for each verb its decomposition into domain predicates and arguments

(3) selectional constraints, which specify the semantic class of arguments which are acceptable in each argument position of each predicate

3.3. Noun Phrase Semantics

Noun phrase semantics has the task of establishing the referent of each noun phrase in the text. More precisely, our system incorporates a model which is isomorphic to the equipment being discussed, and the aim of noun phrase semantics is to replace each noun phrase by the identifier of the thus-named item in the model. Sometimes the noun phrase is not specific enough to identify a single component; in such cases, noun phrase semantics returns the set of possible identifiers (or equivalently, an incompletely specified structure referring to a class of objects in the model). Such references are resolved either during discourse analysis or by a query to the user.

The task of noun phrase semantics is complicated by the prevalence in the CASREPs -- as in many similar technical texts -- of long compound nominals such as "starting air regulating valve" and "SAC [starting air compressor] lube oil alarm". Furthermore, a single part can be referred to by many different names, so it is not feasible to simply have a large dictionary with all the part names. One must determine the structure of the noun phrase, analyze the meaning of the various modifiers, and then locate the part or parts possessing all those properties.

The first step in this process is parsing the noun phrase -- determining its structure. For compound nominals, syntactic categories alone offer almost no guidance in parsing, so the syntactic analyzer passes compound nominals through as unanalyzed strings. Noun phrase semantics uses semantic word categories (stored in the lexicon), and a small grammar stated in terms of these categories, to parse the compound nominal phrase. A single phrase may sometimes have several parses. Second, noun phrase semantics consults the equipment model to determine the elements of the model to which the noun phrase may refer. This process proceeds in essence as follows: associated with some classes of nouns is a list of the elements in the model describable by this noun; for example, associated with "shaft" is a list of all the shafts in the system. Associated with each production in our small noun phrase grammar is a rule which computes the possible referents of the larger phrase from the referents of the constituents. For example, the phrase "compressor shaft" will be analyzed by a production which combines two equipment names ("compressor" and "shaft") to form a single larger equipment name. The corresponding interpretation rule selects those referents of the second equipment name ("shaft") which bear one of a specified set of structural relationships to some referent of the first equipment name ("compressor"). In our domain the relationships include containment and adjacency.

This interpretation process operates compositionally. Thus for the phrase "starting air regulating valve" we first compute the referents for "starting air" (air used for starting something). We then look for valves which perform the function *regulate* on some starting air. The interpretation process may have the side effect of eliminating some parses of the noun phrase: if two constituents related in a parse cannot be related in the model, the analysis is rejected as being meaningless for this piece of equipment. For example, in analyzing the phrase "lube oil pump drive gear", the interpretation process will accept the parse where *lube oil* modifies *pump* (and identify those pumps which pump lube oil), but reject the parse where *lube oil* modifies *gear*. The process of noun phrase analysis is discussed in more detail in (Ksiezyk, Grishman, and

Sterling 1987).

We see from these few examples that noun phrase semantics is heavily dependent upon information from the structure of components (containment, adjacency), the function of components *(regulating* valve), and other properties of components *(high speed* gear box).

3.4. Discourse analysis

When semantic analysis is complete, the text representation consists of domain-specific predicates (conveying the state, change of state, operation, and inspection of equipment) whose operands are, for the most part, references to the domain model. In addition, there are higher-order predicates which are applied to these domain-specific predicates to indicate negation, time relations, belief, ability, etc.

Discourse analysis begins by restructuring this representation into a set of "elementary facts": states, activities, and processes occurring over particular time intervals. Each domain-specific predicate is decomposed into a cluster of such facts. These facts will incorporate information about belief and certainty, derived directly from higher-order predicates, and information about causal and temporal relations developed during discourse analysis.

Determination of these temporal and causal relations is the central function of discourse analysis for these messages. Discourse analysis first extracts the temporal relations conveyed in the message through tense, aspect, and explicit temporal connectives (see (passoneau 1987)). For example, for the message (a slight variant of an actual CASREP):

> DIESEL WAS OPERATING WITH SAC DISENGAGED.
> SAC LUBE OIL ALARM SOUNDED.
> BELIEVE COUPLING FROM DIESEL TO LUBE OIL PUMP TO BE SHEARED.

discourse analysis would determine that the alarm began to sound during the interval when the diesel was operating and the SAC was disengaged. However, it would not be able to determine from the message text alone the time relation between the shearing (sentence 3) and the other events.

This initial set of temporal relations is then augmented through the use of detailed domain information. In particular, discourse analysis relies on causal reasoning to establish the sequence of events. In the example above, it would determine that, in a state where the diesel is operating and the SAC disengaged, shearing the coupling will cause the lube oil alarm to sound. On the other hand, sounding the alarm would not cause the coupling to shear. It would conclude, therefore, that the shearing of the coupling must have preceded the alarm's sounding.

The output of discourse analysis is a set of elementary facts tied together by a rich network of causal and temporal links. The information required for specific message processing applications may be readily extracted from this network. The temporal links may be used to "replay" the events in sequence and thus verify correct message understanding. The initial cause of a failure, if it has been determined, can be identified from the starting point of a causal chain. The effects of a failure can similarly be identified from the endpoint of such a chain.

We see from this description that discourse analysis, like noun phrase analysis, is heavily dependent on domain information. We therefore turn now from the language processing *per se* to the domain model on which it relies.

4. The domain model

In our presentation of the language analyzer we described various types of information about the domain which was required for a full understanding of the messages, including

- structural information about the components (part/whole relations and connections between items at the same level)

- properties of the components (e.g., the shape of gears, the normal operating pressure of fluids)

- functional information about the components (the inputs and outputs of operating components, and the function they perform)

- operational information about the system as a whole: how external inputs and component failure can affect the operation of the system.

4.1. The simulation model

Of the four types of information just mentioned, operational information - which may involve complex relationships between components - is the most challenging to represent. Yet information of this type is clearly needed as part of the causal reasoning in discourse analysis. We considered representing such information directly by a collection of rules, but were concerned that with this approach it would be difficult to anticipate and cover all the possible component failures and their effects. We chose instead to build a *simulation model* of the system: a model which, given the external inputs and the status (OK/damaged) of each component, can compute all the parameters of the systems.

The messages do not in general presume a detailed quantitative knowledge of system behavior, so we chose to construct a *qualitative simulation*, where system parameters take on typically 2 or 3 values. Because the system involves no time-dependent feedback, we could employ a standard discrete-event simulation algorithm. The simulation model is organized in a hierarchical fashion, reflecting the organization of the equipment into identifiable systems and subsystems. For our domain - the starting air system - we have somewhat over 200 operating elements and subsystems.

We have coupled the domain model with a graphical display, in which the various elements and connections are represented by icons, and have animated the display to reflect the state of the system (gears turn, fluids flow, etc.). This combination of simulation and display provides direct feedback to the user entering the message: the system's interpretation of the message - a sequence of events and their effect on the equipment - is made visible to the user.

As we developed the model we recognized the importance of isolating the linguistic analysis from the specific representation chosen for the domain model. To this end we introduced an additional module, *Model Query Processor*. The Model Query Processor accepts queries from the language analyzer stated in terms of the relations and events of the domain, but independent of the model representation. It then translates these queries into tests on specific fields and links in the model. The resulting overall system structure is shown in Figure 1.

4.2. Special features of the model

We have indicated that the needs of our language analyzer for domain information can be effectively satisfied by a simulation model of the domain. For the most part, this is a "conventional" simulation model, comparable to models being used for diagnosis and intelligent computer-aided instruction. However, the task of language analysis does impose some unconventional demands on the model. We mention two of them here.

One basic issue in designing a model is the *level of* detail to be incorporated in the model. The most straightforward approach is to include in our model all the equipment details which may be mentioned in our messages. Unfortunately, our messages sometimes talk about very fine details of the equipment - for example, individual screws, or individual teeth or gears - and incorporating all of these details would create an enormous model. Our approach has instead been to build a static model incorporating larger components such as shafts and gears, and to add finer details (pins and teeth) dynamically if their role in the system is completely determined by the larger element of which they are a part (the tooth of a gear, for example). For a gear or shaft, in contrast, we need to record the adjacent elements to which it is linked in the drive train, so the gear or shaft is made part of the static model.

Dynamic model creation is also needed when a message mentions, as a unit, a combination of elements which does not correspond to any single node in the hierarchical model. For example, one of our messages contains the phrase "believe coupling from diesel to SAC lube oil pump to be sheared". This *coupling* consists of a drive shaft and several gears, and cuts across several natural units in our predefined model. However, we do need to treat it as a unit in some sense, since we wish to associate with it the property of being sheared. We do so by creating, dynamically, a grouping of components in the model corresponding to the noun phrase.

As these examples indicate, the natural language messages deal both with individual objects and groupings which are part of the reader's prior domain knowledge, and with objects and groupings introduced by the narrative. The domain model we use must be capable of dealing with and reasoning about both types of objects, separately and in combination.

5. Discussion

The analysis of discourse structure has been a prime topic of study in computational linguistics for at least the past fifteen years. Common to much of this research has been the recognition of the need for world knowledge in resolving referents and understanding the relations between events. A central issue, therefore, has been the representation and organization of this knowledge. Among the early work in this area was that of Charniak, who developed representation using clusters of inference rules (1972) and elaborate frames (1978), and that of Schank and his colleagues. Schank (Schank and Abelson 1977) introduced *scripts* to account for stereotyped event sequences and *plans* for identifying causal relations in less stereotypical situations.

Although there has been considerable further work in discourse analysis - in plan formalisms, in higher-level discourse structure, and in the effects of discourse structure on anaphoric reference - these basic ideas of scriptal. plan-based, and frame-based knowledge remain at the core of most discourse analysis systems. The basic ideas of plan-based analysis, in particular, are evident in our causal analysis procedure.

However, these basic ideas have been applied for the most part to highly restricted, sometimes artificially constructed, domains. These prior efforts therefore left unanswered many of the questions we faced in the analysis of equipment messages: how to provide adequate coverage of the possible causal relations in a complex system? how to structure the domain model so that it can provide, in an integrated fashion, the information needed by the various stages of language analysis? We believe that our success in using a simulation model as the unified source of most domain knowledge will provide guidance and precedent for further work in the analysis of technical discourse.

6. Does it know too much?

Our approach is based upon a very rich domain model. In our initial focus on understanding the types of domain information required, and how it is to be used in language analysis, we have been less concerned by the large volume of domain knowledge required. However, in the long term this will be a serious concern for both practical and theoretical reasons. The practical concern is that, to be useful, the system must be extended to a wide range of equipment, and this will require the collection of a vast amount of knowledge. The theoretical concern is that we seem to have elaborated in the model more detailed information about the equipment than knowledgeable people require to understand these messages.

Put another way, people probably rely much more on *general* knowledge about equipment, so that they need not remember each detail of each piece of equipment. We have gone a small way toward generalizing our knowledge by introducing into our model *generic* subsystems, with parameters which are assigned different values for different instantiations within the overall system. We need to extend this general knowledge much further; for example, to deduce details of the structure of a subsystem from the function it is expected to perform. In this way we hope to reduce to manageable proportions the knowledge required to understand messages about a broad range of equipment.

7. Acknowledgements

This research was supported in part by the Defense Advanced Research Projects Agency under contract N00014-85-K-0163 from the Office of Naval Research and by the National Science Foundation under grant DCR-85-01843.

This system represents a collaborative effort of the PROTEUS project research group. Past and present members who contributed to this system include: Tomasz Ksiezyk (domain model and discourse analysis), Ngo Thanh Nhan (grammar and clause semantics), John Sterling (noun phrase semantics), Jean Mark Gawron (parser), Leo Joskowicz (discourse analysis), and Michael Moore (Symbolics implementation). We have also collaborated with researchers at UNISYS Defense Systems, Paoli, PA.

8. References

Charniak, Eugene

1972 *Towards a model of children's story comprehension*, Report AI TR-266, Massachusetts Institute of Technology

1978 "On the use of framed knowledge in language comprehension" , *Artificial Intelligence* 11: 225-265

Hobbs, Jerry

1982 "Towards an understanding of coherence in discourse." *Strategies for Natural Language Processing*, edited by W. Lenhert and M. Ringle. Hillsdale, NJ: Lawrence Erlbaum Assoc.

Ksiezyk, Tomasz, Grishman, Ralph, and Sterling, John

1987 An equipment model and its role in the interpretation of noun phrases. *Proc. Tenth Int'l Joint Conf. on Artificial Intelligence*, Milan, Italy

Palmer, Martha

1983 Inference-directed semantic analysis. *Proc. Natl. Conf. Artificial Intelligence (AAAI-83)*

Passoneau, Rebecca

1987 A computational model of the semantics of tense and aspect. *Computational Linguistics*

Sager, Naomi

1981 *Natural Language Information Processing*. Reading, MA: Addison-Wesley.

Sager, Naomi, and Grishman, Ralph

1975 The Restriction Language for Computer Grammars of Natural Language. *Comm. Assn. Computing Machinery* 18: 390.

Schank, Roger, and Abelson, R.

1977 *Scripts, Plans, Goals, and Understanding*. Hillsdale, NJ: Lawrence Erlbaum Assoc.

Thompson, Henry, and Ritchie, Graeme

1984 "Implementing natural language parsers" . *Artificial Intelligence Tools, Techniques, and Applications*, edited by T. O'Shea and M. Eisenstadt. New York: Harper and Row

Wilensky, Robert

1981 "PAM and Micro-PAM". *Inside Computer Understanding*, edited by R. Schank and C. Riesbeck. Hillsdale, NJ: Lawrence Erlbaum Assoc.

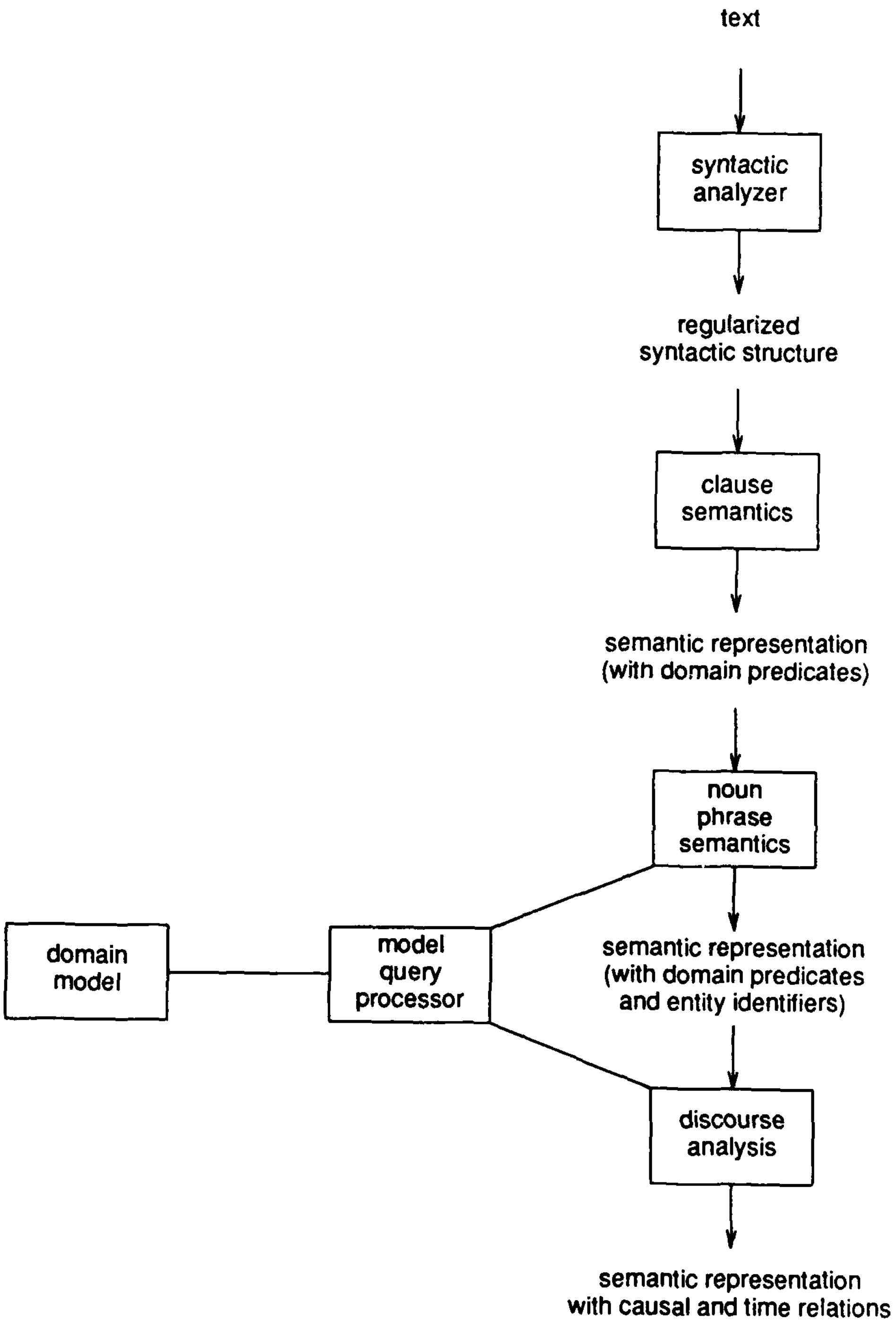
text
syntactic analyzer
regularized syntactic structure
clause semantics
semantic representation (with domain predicates)
noun phrase semantics
semantic representation (with domain predicates and entity identifiers)
domain model
model query processor
discourse analysis
semantic representation with causal and time relations

Figure 1. The principal components and data flow of the PROTEUS system.